Practical-7

AIM: Write a program to implement flow control at data link layer using SLIDING WINDOW PROTOCOL. Simulate the flow of frames from one node to another.

Program should achieve at least below given requirements. You can make it a bidirectional program wherein receiver is sending its data frames with acknowledgement (Piggybacking).

Create a sender program with following features:-

1. Input Window size from the user.
2. Input a Text message from the user.
3. Consider 1 character per frame.
4. Create a frame with following fields [Frame no., DATA].
5. Send the frames. [Print the output on screen and save it in a file called Sender_Buffer.]
6. Wait for the acknowledgement from the Receiver. [Induce delay in the program]
7. Reader a file called Receiver_Buffer.
8. Check ACK field for the Acknowledgement number.
9. If the Acknowledgement number is as expected, send new set of frames accordingly, [overwrite the Sender_Buffer file with new frames] Else if NACK is received, resend the frames accordingly. [Overwrite the Sender_Buffer with old frame].

Create a receiver file with following features

1. Reader a file called Sender_Buffer.
2. Check the Frame no.
3. If the Fame no. are as expected, write the appropriate ACK no. in the Receiver_ Buffer file. Else write NACK no. in the Receiver_Buffer file.

NOTE: Induce error and verify the behaviour of the program. Manually Change the Frame no and Ack no in the files].

CODE:

Sender.py

```
import time
window_size = int(input("Enter window size: "))
message = input("Enter text message: ")
frames = [[i+1, message[i]] for i in range(len(message))]
base = 0
while base < len(frames):
   end = min(base + window_size, len(frames))
   sender_buffer = ""
   for i in range(base, end):
      sender_buffer += f"[Frame:{frames[i][0]},Data:{frames[i][1]}]\n"
   open("Sender_Buffer.txt", "w").write(sender_buffer)
   print("Sent Frames:")
   print(sender_buffer)
   time.sleep(2)
   ack_data = open("Receiver_Buffer.txt").read().strip()
   print("Receiver Response:\n", ack_data)
   if "ACK" in ack_data:
      ack_no = int(ack_data.split(':')[1])
```

```
        if ack_no == end:
            print("All frames acknowledged.\n")
            base = end
        else:
            print("Partial ACK received, sliding window accordingly.\n")
            base = ack_no
    elif "NACK" in ack_data:
        nack_no = int(ack_data.split(':')[1])
        print("NACK received for Frame", nack_no, "- Resending...\n")
        base = nack_no - 1
    else:
        print("No proper ACK/NACK, resending same frames.\n")
```

Receiver.py

```
import time
data = open("Sender_Buffer.txt").read().strip().split('\n')
receiver_buffer = ""
expected_frame = 1
for line in data:
    if line.strip():
        num = int(line.split(',')[0].split(':')[1])
        ch = line.split(',')[1].split(':')[1].replace(']', '')
        if num == expected_frame:
            print(f"Received Frame {num} with data '{ch}'")
            expected_frame += 1
        else:
            print(f"Frame {num} unexpected. Sending NACK {expected_frame}")
            open("Receiver_Buffer.txt", "w").write(f"NACK:{expected_frame}")
            time.sleep(1)
            exit()
ack_data = f"ACK:{expected_frame - 1}"
open("Receiver_Buffer.txt", "w").write(ack_data)
print("All frames received correctly. Sending", ack_data)
time.sleep(1)
```

Input(sender):

```
Enter window size: 3
Enter text message: HELLO
```

Output(sender):

```
Sent Frames:
[Frame:1,Data:H]
[Frame:2,Data:E]
[Frame:3,Data:L]

Receiver Response:
ACK:3
All frames acknowledged.
Sent Frames:
[Frame:4,Data:L]
[Frame:5,Data:O]
```

Practical-7

```
Receiver Response:
ACK:5
All frames acknowledged.
```

Receiver(output):

```
Received Frame 1 with data 'H'
Received Frame 2 with data 'E'
Received Frame 3 with data 'L'
All frames received correctly. Sending ACK:3
Received Frame 4 with data 'L'
Received Frame 5 with data 'O'
All frames received correctly. Sending ACK:5
```